A Depth Analytical Re-appraisal of Efficient Load equilibrium in Cloud Environment

Synopsis submitted to the IIMT University in partial fulfillment of the requirements for the award of the Degree of

DOCTOR OF PHILOSOPHY

In

Computer Science

By

Anjali Sharma

Under the guidance of

Dr. Prof.Garima Sinha

IIMT Group of College

IIMT University, Meerut

School of Computer Science and Applications

Introduction

- The development of apps and the utilization of the current system to create new software are all aspects of the cloud computing technology. The goal of cloud computing is to offer computing as well as other services like power, water, gas, and phone bills. Users can access these services as needed, and they are paid based on how much time they spend using them. Pay-per-usage of services is the name of this pricing structure.
- Many benefits of cloud computing include cost savings for IT services, flexibility, and scalability. While though cloud computing has many benefits for many services, it also has drawbacks, including those related to security, privacy, and response time, among others. One of the problems with inappropriate load balancing in the cloud is response time, which makes it difficult for the cloud service provider to meet customers' needs effectively. While balancing the incoming load on the cloud server, the majority of the reviewed algorithms—including Min-Min, Max-Min, and Ant Colony Optimization—have a problem with response time.

Objective

- The aim of research work is to design an algorithm for efficient load balancing system on Cloud Computing. Which work on response time, throughput and overheads related to Virtual Machines (VM's).
- To analyze load rebalancing approach for reducing load migration time and improvement of response time.
- To prioritize Cloud customer based on service demand and geographical location for better customer satisfaction.
- To improve the resource utilization in a Cloud system using better load balancing techniques.

Literature review

D. Chitra Devi and V. Rhymend Uthariaraj talk about how to beat matters related to round-robin load-balancing algorithm and give their suggestions on that how to minimize this over burden on nodes. He discussed the weighted round- robin plan in which whole turnaround time for each user job is enhanced.

Sudha Senthil Kumar et al explained enhanced report based on the Honey bee foraging Algorithm. They did the calculation of adaptive neighboring search and site abandonment strategy with Honey bee foraging

algorithm. And efficiency is improved by this approach.

Carrera et al have look at the main problem of supervision the heterogeneous workloads in virtualized data centers. They have given a technique which work dynamically and modifies the heterogeneous workload placement on the similar physical hardware. The main advantage of these functions is used in this technique which helped them and also compares the performance of diverse workloads. These proposed ideas better the performance of heterogeneous workloads.

Xu et al have explored a lot of load balancing algorithms for virtual machine assignment in Cloud environment. They have also explored the number of challenges like overhead, performance, etc. related to load balancing in the virtual environment. As a future scope, they have talked about the designing of self-adaptive dynamic load balancing algorithm.

Chen proposed an algorithm known as user based Priority Aware Load Balanced Improved Min-Min (PA-LBIMM) scheduling algorithm that utilizes features min-min algorithm, the basis to improve load balancing factor by minimizing the end time of all resources. Here the process submitted by the user is categories into two sections, ordinary and the VIP sections such that VIP tasks are executed first by getting the VIP resources. Once the VIP tasks assignment completes, the ordinary tasks assignment starts and remaining VIP and ordinary resources are allotted to them. Once all task assignment completed, and then load balancing will start by picking the task with smallest amount end time from the large amount loaded resource. Minimum completion times of that resource is also considered on extra resources and check its least amount completion time. But the make-span of that task with those resources having a greater value than the minimum response time then the particular task is reassigned to that resource. This will continue till the most heavily loaded resources have no other task with minimum finish time to be rescheduled. As we know that pay by the cloud as per usage model, so assigning the responsibilities which based on user main concern is a good quality idea for scheduling the weight.

Ant Colony Optimization Load balancing algorithm is projected in that will utilize the basic Ant Colony Optimization algorithm give an idea to balance the load in the Cloud based environment. This algorithm will reduce the make-span on a given task to balance the load. In this algorithm initially, the pheromone is allocated based several processors, their capacity and bandwidth and then place the ant on VM's randomly. Next, best VM is chosen by getting the chances which is calculated as the ration of the pheromone of that machine to the ration of the pheromone of all machines. Best VM for remaining ant is selected from all the neighboring VM based on the maximum probability value of all the VM's that are calculated. Pheromone table is updated locally once an ant completes the tour. If the answer attained is the current best answer, then

the global modernize of pheromone in the table is completed. This scheduling algorithm carry on until or unless to iterative condition satisfies. All the ants complete their tour for each iteration. The results will be generated using CloudSim simulator and compared with FCFS and ACO based on load imbalance and makespan of the tasks. Results show this algorithm performs better than FCFS and ACO.

Ghafari proposed the Artificial Bee Colony algorithm –Minimal Migration Time (Bee-MMT) that utilizes features of bee colony algorithm to find the overloaded host and then MMT selection procedure is used for virtual machines(VM's) to decide which VM need to be shifted from the overloaded host. It also recognizes the under- loaded hosts and, then migrate all the VM's assigned to them to the other suitable host. Once the under-loaded host becomes free it is switched off to save the energy. The proposed approach is used to solve the problem of power consumption concerning to the response time of VM. This approach is also not one of the best methods in green Computing concepts.

Panwar et.al (2015) suggested and given a proposed balancing algorithm which also work on load by allocate the dynamic load in the regularly on the servers by the help of investigation of the recent status of every potential VMs and also suggested the amount the response time of their proposed algorithm.

Kapur et.al (2015) projected an algorithm which based on load balancing approach for cloud computing called Load Balanced Resource Scheduling Algorithm (LBRS) that allowing two categories of resources: one is worked on reservation resources and another one is based On-Demand resources.

A. Kaur and B Kaur (2019) planned an idea which based on a framework for VM load balancing approach and the name is, ''Hybrid approach based Deadline-constrained, Dynamic VM Provisioning and Load Balancing (HDD-PLB)". This framework performance has been investigated on the two important bases which called the metrics. One important metrics is Makespan and another one is Cost metrics.

Methodology

In this research proposal a different Methodology are used related to good Load Balancing Algorithms. Like

- **Load Estimation Policy**

Load Estimation policies are used to define the total workload on a node in a system. The load can be estimated based on the total number of processors on the node, resource demands, architecture, and speed of the nodes processor and instruction mixes of these processes. Many load balancing algorithms use the total

number of processes to achieve that degree of efficiency. In some circumstances, the genuine load could vary substantially based on the remaining service time of processes on a node, which can be measured in different ways such as Memory less technique, past repeats and Distribution method. A system that measures CPU consumption of nodes is commonly used for load estimation. CPU utilization can be measured by the CPU states (busy/idle) based on setting up the timer that from time to time checks.

- **Process Transfer Policy**

It selects the processor for transferring the load which would increase the throughput and response time. The idea is to transfer processes from loaded nodes to less loaded nodes. Threshold policy is utilized by most of the algorithms to decide lightly or heavily loaded. Limit of workload on load value is decided by threshold value that is determined using:

***Static policy*:** The present threshold value for each node depends on the processing capacity of CPU in the static policy.

***Dynamic policy*:** In dynamic policy threshold workload on each node is calculated using average load value and the predefined constants.

If the node having values less than the threshold, then it accepts the processes to execute and if the value exceeds the threshold, the processes are transferred to a lightly loaded node. Algorithms with single threshold policy may lead to unstable conditions because once the process is migrated to lightly loaded node it might become heavily loaded. Double threshold policy is also called high low policy, used to reduce instability issue.

- **State Information Exchange Policy**

It gathers workload-related information about the system, such as the workload's nature and the average load on each node. For dynamic policies repeated exchange of state, information is a must. This exchange of information occurs through messages. These extra messages result in two impacts, one it provides more accurate results and another more figure of communication boost queuing instances of communication. Situation information strategies that can be generally used are:

***Periodic broadcast*:** In periodic broadcast, state information is broadcast by every node after every T units of time. Some of the problems occur in this type of broadcast is related to loaded traffic and reduced

scalability.

***Broadcast when the state changes*:** In this method, broadcast happen only when processes arrive or depart from the node. It means broadcasting is done only when the state changes.

***On-demand exchange*:** this method when a node condition has changed from normal to overload or under load condition, then State-Information-Request message is broadcast. To the querying node, all nodes respond with their status information. Once they get the broadcast message. Steps were taken to make this request information more relevant. For example, the nodes which are useful (or having useful information) will reply to the requesting node.

***Exchange by polling*:** The polling mechanism overcomes the low scalability of broadcast messages. There is no need for a node to share its status information. The partner node is found by polling the other nodes one at a time until the poll limit is met.

- **Priority Assignment Policy**

A priority assignment rule for scheduling a node's local and remote processes should be planned in a distributed operating system that enables process migration. One of the priority assignment rules listed below may be used:

- By rule of Selfish priority assignment
- By rule of Altruistic priority assignment
- By rule of Intermediate priority assignment
- By rule of Migration limiting policy for Load-balancing algorithms

Uncontrolled: A remote process incoming at a node is treating presently as a process initiate at a node due to which a process may be traveled any amount of times.

Controlled: This employs a migration calculate parameter to put a limit on how many times a process can migrate, avoiding the volatility of the uncontrolled policy. Compared to a short process, a extended process may be permitted to migrate more regularly.

Conclusion

- This Synopsis is set to introduce Cloud Computing in conjunction with the background of various load balancing techniques to balance the workload in the Cloud server. The research started with the acmes of virtualization technology, its applications and the advantages procured through Cloud Computing. At the

outset the thesis discusses the importance and necessity of Cloud Computing for IT organizations focusing on the matrices like throughput, response time, etc. needed to improve the efficiency of the Cloud. Ensuing to this importance and necessity of Cloud Computing for IT organizations, the thesis concentrates on the different load balancing techniques used in the Cloud which are basically of two types: Static and Dynamic.

- Static load balancing techniques are good for systems where there are initial information of basic load and the configurations of the systems. Some of static load balancing techniques discussed here in the thesis are round-robin, min-min, max-min, etc. All these algorithms have some limitations associated with response time as they are unable to tackle real-time load variations.
- Dynamic load balancing is used for heterogeneous load. Here the load balancer maintains all the information about the VM's or DC at run time. Some of dynamic load balancing techniques are Ant colony optimization and honey bee forging techniques. Though these techniques also have issues allied to throughput, response time etc., they are capable of asphyxiating some of the issues associated with static techniques.

Bibliography or References

[1] D. Chitra Devi and V. Rhymend Uthariaraj, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks", The Scientific World Journal, Vol 2016, Article ID 3896065, 14 pages

[2] Sudha Senthil kumar, K.Brindha, Rathi, Angulakshmi, Jothi and Yash Vardhan Thirani (2017), ―Honey-Bee Foraging Algorithm for Load Balancing in Cloud Computing Optimization", International Journal of Engineering Science and Computing, Vol. 7, Issue No. 12, pp. 15840- 15844.

[3] D. Carrera, M. Steinder, I. Whalley, J. Torres and E. Ayguade, "Managing SLAs of heterogeneous workloads using dynamic application placement", HPDC '08 Proceedings of the 17th international symposium on High performance distributed Computing, New York, USA, 2008.

[4] Minxian Xu Wenhong Tian, Rajkumar Buyya1, "A survey on load balancing algorithms for virtual machines placement in Cloud Computing", Wiley, Feb 2017, pp. 1-16.

[5] Huankai Chen; Wang, F.; Helian, N.; Akanmu, G., "Userpriority guided Min-Minscheduling algorithm for load balancing in Cloud Computing," Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on , vol., no., pp.1,8, 21-23 Feb. 2013.

[6] Kun Li; Gaochao Xu; Guangyu Zhao; Yushuang Dong; Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization," Chinagrid Conference (ChinaGrid), 2011 Sixth Annual ,

vol., no., pp.3,9, 22-23 Aug. 2011.

[7] Ghafari, S.M.; Fazeli, M.; Patooghy, A.; Rikhtechi, L., "Bee- MMT: A load

[8] Panwar, R., Mallick, B., 2015, October. 'Load balancing in cloud computing using dynamic load management algorithm', IEEE International Conference on Green

[9] H. Ali, I.A. Saroit, A.M. Kotb, "Grouped tasks scheduling algorithm based on QoS in cloud computing network", Egyptian Informatics Journal, July 2017, pp 11-19.

[10] J. Srinivas, K. Venkta Subha Ready, Dr. A. Moiz Qyser, " Cloud Computing Basics", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 5, July 2012, pp 343-346.